Defining the Voice Interface

Introduction

With the Alexa Skills Kit, you can add new capabilities to Alexa by creating *custom skills*. In addition to implementing the logic for a new custom skill, you also define the *voice interface* your users will have with the skill, and through which they experience the new capability.

You define this voice interface by specifying a mapping between users' *spoken input* and the *intents* your cloud-based service can handle.

To declare this mapping, you supply two main inputs:

1. An **Intent Schema**: A JSON structure which declares the set of intents your service can accept and process.
2. The **Spoken Input Data:**
    - Sample Utterances: A structured text file that connects the intents to likely spoken phrases and containing as many representative phrases as possible.
    - Custom Values (required for use with custom slots): A representative list of values for specific items used by your skill and referenced in the intents when using a custom slot type.

You enter this information in the developer portal on the **Interaction Model** page. For details, see Registering and Managing Custom Skills in the Developer Portal.

The following sections explain how to use each of these resources to define an effective voice interface.

The Intent Schema

Alexa Skills Kit - Custom Skills

- Understanding Custom Skills
- Get Started with Sample Code
- Voice Design
    - **Define the Voice User Interface**
    - Understanding How Users Invoke Custom Skills
    - Choosing the Invocation Name for a Custom Skill
    - Voice Design Handbook
    - Voice Design Best Practices
- Setting Up
- Developing
- Displaying Skill Cards
- Testing
- Publishing
- Reference

Other Resources

- Getting Started
- Smart Home Skills
- Flash Briefing Skills
- Glossary
- Alexa Skills Kit Forum
- Alexa Voice Service
- Alexa Fund

In the context of Alexa, an *intent* represents an action that fulfills a user's spoken request. Intents can optionally have arguments called *slots*. For example, the intent schema for "Daily Horoscopes" might define an intent named `GetHoroscope` that contains a slot named `Sign`. When a user says "Alexa, ask Daily Horoscopes for the horoscope for Pisces", the Alexa service sends the Daily Horoscopes service a `GetHoroscope` intent with the value "pisces" in the `Sign` slot. On receiving the `GetHoroscope` intent, the service can then look up the horoscope for Pisces and send back text to convert to speech.

You define the set of valid intents in a JSON structure called an *intent schema*. For example, the following intent schema defines two intents: `GetHoroscope` and `GetLuckyNumbers`:

```
{
  "intents": [
    {
      "intent": "GetHoroscope",
      "slots": [
        {
          "name": "Sign",
          "type": "LIST_OF_SIGNS"
        },
        {
          "name": "Date",
          "type": "AMAZON.DATE"
        }
      ]
    },
    {
      "intent": "GetLuckyNumbers"
    }
  ]
}
```

Each intent has two properties:

- The `intent` property gives the name of the intent.
- The `slots` property lists the slots associated with that intent.

In the above example, the `GetHoroscope` intent defines two slots, one named `Sign`, the other named `Date`.

Slots are defined with different *types*. For instance, the `Sign` slot shown in the above example is defined as a custom type `LIST_OF_SIGNS` to reference a list of horoscope sign values provided for the slot (Aries, Capricorn, etc.), and the `Date` slot would convert a month name spoken by the user into a date format using Amazon's built-in AMAZON.DATE type.

Note that the Alexa Skills Kit provides a library of built-in intents for common actions. If you want to implement these, you include them in your intent schema just like your own custom intents. For details, see Implementing the Built-in Intents.

The Speech Input Data

You provide the mappings between the intents and the typical utterances that invoke those intents by adding sets of values for any custom slots supported by your skill and a list of sample utterances.

Built-in Slot Types

Amazon provides built-in support for a large library of slot types. This includes types for converting data such as dates, and types that provide recognition for commonly-used lists of values, such as cities in different countries, regions and states, and first names.

For example, the AMAZON.DATE type converts words that indicate *dates* ("today", "tomorrow", or "july") into a date format (such as "2015-07-00T9").

> **Tip:** For the complete list of all built-in slot types and examples, see Slot Type Reference.

If your skill uses a built-in type, you don't need to provide sample values. Include the slot name in the sample utterances file as necessary. For other data types, a custom slot type is recommended. Note that all built-in types are prefixed with the `AMAZON` namespace.

An additional built-in slot type is supported just for *English (US)* skills: `AMAZON.LITERAL`. This type is primarily provided for compatibility with earlier versions of the Alexa Skills Kit.

> **Note:** Based on developer feedback, the `AMAZON.LITERAL` slot is not being removed as previously described. You can continue to submit new and updated English (US) skills with `AMAZON.LITERAL`. However, in many cases, custom slot types provide better accuracy than `AMAZON.LITERAL`, so we recommend that you consider migrating to custom slot types if possible.
>
> English (UK) and German skills do not support `AMAZON.LITERAL` and cannot use the `AMAZON.LITERAL` slot type.

Note that slot type behavior has changed since earlier versions of the Alexa Skills Kit. For information about the differences, backward compatibility, and migrating a skill from the previous version, see Migrating to the Improved Built-in and Custom Slot Types.

Custom Slot Types

A custom slot type is used for items that are not covered by Amazon's built-in set of types and is recommended for most use cases where a slot value is one of a set of possible values. For the `LIST_OF_SIGNS` slot type in the horoscope sample above, the list of custom values would be:

```
Aries
Taurus
Gemini
Cancer
Leo
Pisces
Virgo
Libra
Scorpio
Sagittarius
Capricorn
Aquarius
```

The set of custom values can be anything supported by your skill's handling of the slot as long as it can be spoken by a user, although words not found in a typical English dictionary may not be recognized.

Slot values are sent to your skill in written format. For example, both "fire h.d. 7" and "fire h.d. seven" would be sent to your skill as "Fire HD7". For better recognition, acronyms and other phrases involving spoken letters should either be all caps ("HD") or separated by periods ("h.d."). Using lowercase for initialisms may lead to unreliable recognition since the spoken form may not correctly be detected. For examples, see the "Custom Slot Type Syntax" section of the Custom Interaction Model Reference.

For recommendations for custom slot type values, see Custom Slot Values, below.

A custom slot type is not the equivalent of an enumeration. Values outside the list *may still be returned* if recognized by the spoken language understanding system. Although input to a custom slot

type is *weighted* towards the values in the list, it is *not constrained* to just the items on the list. Your code still needs to include validation and error checking when using slot values. See the "Handling Possible Input Errors" section of Handling Requests Sent by Alexa.

A custom slot type can be used for multiple slots. For example, suppose the horoscope skill had an additional intent, `MatchSign`, that compared two Zodiac signs. This intent defines two slots, `FirstSign` and `SecondSign`. These slots can both use the same custom `LIST_OF_SIGNS` slot type:

```
{
  "intents": [
    {
      "intent": "GetHoroscope",
      "slots": [
        {
          "name": "Sign",
          "type": "LIST_OF_SIGNS"
        },
        {
          "name": "Date",
          "type": "AMAZON.DATE"
        }
      ]
    },
    {
      "intent": "MatchSign",
      "slots": [
        {
          "name": "FirstSign",
          "type": "LIST_OF_SIGNS"
        },
        {
          "name": "SecondSign",
          "type": "LIST_OF_SIGNS"
        }
      ]
    },
    {
      "intent": "GetLuckyNumbers"
    }
  ]
}
```

The Sample Utterances File

Each possible sample utterance is assigned to one of the defined intents. For example, the following snippet from a sample utterances file maps four possible phrases to the `GetHoroscope` intent, two phrases to the `GetLuckyNumbers` intent, and two phrases to the `MatchSign` intent (in practice, many more utterances are required than shown in this example):

```
GetHoroscope what is the horoscope for {Sign}
GetHoroscope what will the horoscope for {Sign} be on {Date}
GetHoroscope get me my horoscope
GetHoroscope {Sign}
...
GetLuckyNumbers what are my lucky numbers
GetLuckyNumbers tell me my lucky numbers
...
MatchSign do {FirstSign} and {SecondSign} get along
MatchSign what is the relationship between {FirstSign} and {Secor
```

Note that you do not need to provide sample utterances for any of the built-in intents, such as `AMAZON.HelpIntent`.

Sample Utterance Syntax

Each line of a sample utterances file consists of two fields separated by tabs or spaces:

- The name of the intent on the left.
- The phrase a user might speak to signal that intent on the right.

Slots are arguments to intents. In each sample utterance, for all slot types except `AMAZON.LITERAL`, represent a slot value as the slot

name in curly braces: {SlotName}. The `AMAZON.LITERAL` slot type also requires word values for the slot inline: {slot value|SlotName}.

For example, consider this sample in the file:

```
"what will the horoscope for {Sign} be on {Date}"
```

In this utterance, the slots {Sign} and {Date} serve as arguments to the `GetHoroscope` intent, and are placed as variables in this utterance pattern. Possible values for {Sign} are taken from the custom list `LIST_OF_SIGNS` and possible values for {Date} are taken from the Alexa Skill Kit's built-in support for dates. This allows the user to give input such as:

```
"what will the horoscope for Leo be on Tuesday"
```

The Alexa Skills Kit automatically generalizes input based on the custom and built-in slot types so just the specification of one pattern allows all forms of input against that pattern. However, it's still important to consider variations in the utterance patterns. For example, to cover the following utterance:

```
"what will the horoscope for libra be tomorrow"
```

the following pattern should be added to the sample utterances file (where "on" is no longer necessary):

```
"what will the horoscope for {Sign} be {Date}".
```

For more guidance on specifying utterance patterns, see the section *Sample Utterances Contents* below.

An individual utterance does not need to use every slot defined for an intent. The schema for the `GetHoroscope` intent shown in the above example defines both a `Sign` slot and a `Date` slot. The utterance "`what is the horoscope for {Sign}`" only uses the `Sign` slot, while the utterance "`get me my horoscope`" doesn't use any slots at all. When users speak these phrases, the same intent is sent, but with `null` slot values.

As a general practice, look out for typos and spelling errors. Don't forget the apostrophe (') in "what's" and "who's", and watch out for mistakes introduced by directly copying and pasting from written language sources.

Sample Utterance Contents

Given the flexibility and variation of spoken language in the real world, there will often be many different ways to express the same request. For example, to ask for a horoscope a user might say:

- what is the horoscope
- get me my horoscope
- tell me the horoscope
- how's my horoscope today
- …

Or any other variations on the above forms:

- "what's" and "what is"
- "get", "tell", and "give"
- "my" and "the"

If you have requests that are full sentences, think about shortened ways that users might say them, especially when combined with the Ask/Tell starting phrases, with patterns like *Ask…to…* and *Ask… about…*. For instance, if you have "what's the weather," consider also just "weather". This works well with patterns such as "Ask <invocation name> for *weather*." or "Ask <invocation name> about *weather*".

Providing these different phrases in your sample utterances will help improve voice recognition for the abilities you add to Alexa. It is important to include as wide a range of representative samples as you can — all the phrases that you can think of that are possible in use (though do not include samples that users will never speak). Alexa also attempts to generalize based on the samples you provide to interpret spoken phrases that differ in minor ways from the samples specified.

Recommendations for Defining the Speech Input Data

The usability of the skill directly depends on how well the custom slot data and sample utterances represent *real-world language use*. Building a representative set of custom values and sample utterances is an important process and one that requires iteration. During development and testing, try using many different phrases to invoke each intent. If you can observe other users during testing, note the phrases that they speak to invoke each intent. Continually update the custom values and sample utterances file to ensure that it includes instances of your users' most common phrasings.

The following sections provide recommendations for developing your set of sample utterances.

Sample Utterances for Starting a Conversation

The preferred phrases for beginning an interaction with an Alexa ability are *ask* and *tell*. These can be used to ask questions or state commands in a very natural way. Your sample utterances should flow naturally when combined with these phrases.

For example, users might say one of the following to request a horoscope:

- "Alexa, **ask** Daily Horoscopes **for** *the horoscope for Gemini*"
- "Alexa, **ask** Daily Horoscopes **about** *Gemini*"
- "Alexa, **ask** Daily Horoscopes *what is the horoscope for Gemini*"
- "Alexa, **ask** Daily Horoscopes *what's the horoscope for Gemini*"
- "Alexa, **ask** Daily Horoscopes **to** *give me the horoscope for Gemini*"
- "Alexa, **ask** Daily Horoscopes **to** *tell me the horoscope for Gemini*"

Note the different forms of the sample utterances that work effectively with different variations of "ask":

- *Noun* utterances:
  - "the horoscope for…"
  - "Gemini"
- *Question* utterances:
  - "what is the horoscope for…"

        - "what's the horoscope for…"
    - *Verb* utterances:
        - "give me the horoscope for"
        - "tell me the horoscope for…"

In the syntax of the sample utterances, these would be specified as follows:

```
GetHoroscope the horoscope for {Sign}
GetHoroscope {Sign}
GetHoroscope what's the horoscope for {Sign}
GetHoroscope what is the horoscope for {Sign}
GetHoroscope give me the horoscope for {Sign}
GetHoroscope tell me the horoscope for {Sign}
...
```

Utterances that begin with specific question words ("what", "how", "where", and so on) are especially important, as these work with the "**ask** <invocation name> *question*" variation. The following words are considered question words that work with **ask**:

- who, what, why, when, where, which, how
- do, did, does
- can, could, should, would, may, must, shall
- is, was, will, am, are, were
- have, has, had
- Some variants of the above, such as: what's, whose, didn't, ain't

When writing utterances as questions, also consider multiple forms of the question. For example:

- "what is my horoscope". ("Alexa, **ask** Daily Horoscopes *what is my horoscope*")
- "what my horoscope is" ("Alexa, **ask** Daily Horoscopes *what my horoscope is*")

Not all of these question words may work well for all Alexa capabilities, but you should include utterances for these words when they represent real-world language use. For the full list of phrases users can say to begin interacting with your Alexa skill, see Understanding How Users Invoke Custom Skills. Include utterances that work *naturally* with as many of these phrases as possible.

Number of Sample Utterances

For each intent, include *as many variations* of the phrases as you expect users to speak. For example, for the utterance "what is my horoscope", include variations such as:

- "what is…"
- "what's…"
- "tell me…"
- "give…"
- "give me…"
- "get…"
- "get me…"
- "find…"
- "find me…"

It is better to provide too many samples than to provide too few, so test different phrases and add additional phrases as needed.

Custom Slot Values

When using custom slot types, make sure the list of values provided for the slot comprehensively covers the expected input from your users. While this is relatively straightforward in the horoscopes example — in this case the set of twelve signs is small, finite and has few or no variations in terms of how people are likely to speak each sign — it can be more difficult as the complexity of input to a given slot increases. Tips to build representative coverage for more complex slot types include:

- Start with data sources that reflect real usage. For instance, if you have a web version of your application that can use items you've logged from users via forms or search queries, use those. Otherwise look for other data sources representing likelihood of input.

  If you're building a list of ingredients for a recipe skill for instance, generate a list of all ingredients used in all recipes, and validate that against other lists you may be able to find online of common ingredients.

- Your skill can have a total of 50,000 custom slot values. This is totaled across all custom slots used in the interaction model. If you need to scale back the number of values to fit within this limit, be sure to keep the values most likely to be used.
- If your list of custom values does not contain *all* the values your skill expects, provide a representative set of custom values with representative word counts. For instance, if values of one to four words are possible, use values of one to four words in your value list. But also be sure to distribute them proportionally. If a four-word value occurs in an estimated 10% of inputs, then include four-word values only in 10% of the values in your list.

Next Steps

- Next: Understanding How Users Invoke Custom Skills
- Return to: Steps to Build a Custom Skill
- Also see: Custom Interaction Model Reference

Alexa
Alexa Skills Kit
Alexa Voice Service
Alexa Fund

Services & APIs
Earn
Engage
Build

Devices
Fire Tablets
Amazon Fire TV
Dash Replenishment Service
Fire Phone
Amazon Echo
Amazon Tap

Resources
Platforms
Learning Center
Development Tools
Promotional Tools
Marketing Tips
Other Resources

Blog
Subscribe to the Blog
#Announcements
#AmazonUnderground
#Alexa
#AmazonFireTV
#FireTablets
#How To
#Monetization
#APIs
#Marketing
#Gamedev
#Developer Spotlight

Support
Submitting Your Apps
FAQs
Forums
Contact Us
App Distribution Agreement
Mobile Ad Network Publisher Agreement
Mobile Ad Network Program Participation Requirements
Advertise Your App With Amazon Agreement
Program Materials License Agreement
Trademark Guidelines
Terms of Use